Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US04/040528

International filing date: 03 December 2004 (03.12.2004)

Document type: Certified copy of priority document

Document details:
Country/Office: US
Number: 60/527,789
Filing date: 05 December 2003 (05.12.2003)

Date of receipt at the International Bureau: 31 March 2005 (31.03.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

1296717

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

March 16, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: ***60/527,789***
FILING DATE: ***December 05, 2003***
RELATED PCT APPLICATION NUMBER: ***PCT/US04/40528***

UNITED STATES PATENT AND TRADEMARK OFFICE
DEPARTMENT OF COMMERCE

Certified by

Jon W. Dudas

Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

PTO/SB/16 (08-03)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Express Mail Label No. EL984296125US

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Yun-Qing Zhicheng	Shi Ni	Millburn, NJ Kearny, NJ

Additional inventors are being named on the 1 *separately numbered sheets attached hereto*

TITLE OF THE INVENTION (500 characters max)

METHOD AND APPARATUS FOR LOSSLESS IMAGE DATA HIDING IN THE PIXEL DOMAIN

Direct all correspondence to: CORRESPONDENCE ADDRESS

☑ Customer Number: 27538

PATENT TRADEMARK OFFICE

OR

□ Firm or Individual Name					
Address					
Address					
City		State		Zip	
Country		Telephone		Fax	

ENCLOSED APPLICATION PARTS *(check all that apply)*

☑ Specification *Number of Pages* 34

□ Drawing(s) *Number of Sheets* ______

□ Application Date Sheet. See 37 CFR 1.76

□ CD(s), Number ______

□ Other (specify) ______

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT

☑ Applicant claims small entity status. See 37 CFR 1.27.

☑ A check or money order is enclosed to cover the filing fees.

☑ The Director is herby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 11-0223

□ Payment by credit card. Form PTO-2038 is attached.

FILING FEE Amount ($): $80.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☑ No.

□ Yes, the name of the U.S. Government agency and the Government contract number are: ______

[Page 1 of 2]

Respectfully submitted,

SIGNATURE ______

TYPED or PRINTED NAME Matthew B. Dernier

TELEPHONE (732) 634-7634

Date ______

REGISTRATION NO. 40,989
(if appropriate)

Docket Number: 436/9

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Mail Stop Provisional Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PROVISIONAL APPLICATION COVER SHEET
Additional Page

PTO/SB/16 (08-03)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Docket Number 436/9

INVENTOR(S)/APPLICANT(S)

Given Name (first and middle [if any])	Family or Surname	Residence (City and either State or Foreign Country)
Nirwan	Ansari	Montville, NJ

[Page 2 of 2]

Number __________ of__________

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

TITLE OF THE INVENTION

Method and Apparatus for Lossless Image Data Hiding In The Pixel Domain

Background

This invention relates to image data hiding that can recover the original images without any distortion after the hidden data have been retrieved from the stego-images. Furthermore, it does not generate salt-pepper noise and is robust against compression.

Data hiding is referred to as a process to embed useful data (information) into a cover media. It has wide applications for the purpose of identification, annotation, copyright protection and authentication. In these applications, people do care about the cover media. That is, the hidden data and the cover media are closely related. This type of data hiding is often referred to as watermarking.

For this type of data hiding, the hidden data are required to be perceptually transparent. In other words, we require the marked media to be as similar to the cover media as possible.

A discussion of lossless data hiding will now be provided. In most of cases, the cover media will experience some distortion due to data hiding and cannot be inverted back to the original media data even after the hidden data have been extracted out.. That is, some permanent distortion exists even after the hidden data have been extracted. For instance, round-off error, truncation error and quantization error make lossless data hiding impossible. However, for some applications such as medical diagnosis and law enforcement, it is imminent to invert the marked media back to the original cover media after the hidden data have been retrieved. The marking techniques satisfying this requirement are referred to as *lossless*, sometimes as *distortionless*. They are also called *reversible* marking techniques. This type of marking techniques is suitable for applications where the exact original media data should be recovered.

Recently, some lossless marking techniques have been reported in the literature. The first method [honsinger 99] is carried out in the image spatial domain. Another spatial domain technique was reported in [fridrich 01]. There also exists a distortionless marking technique in the transform domain [macq 99]. From our study of the transform domain method, the upper bound of the amount of hidden data is estimated to be 2000 bits (equivalent to 250 bytes) for a $512 \times 512 \times 8$ image. Hence, these techniques are not suitable for applications where a much larger amount of data is requested to hide in images. The capacity of the method reported in [vleeschouwer 01] is also very limited except that it exhibits robustness against high quality JPEG compression. These techniques aim at authentication, instead of data hiding. As a result, the amount of hidden data is quite limited.

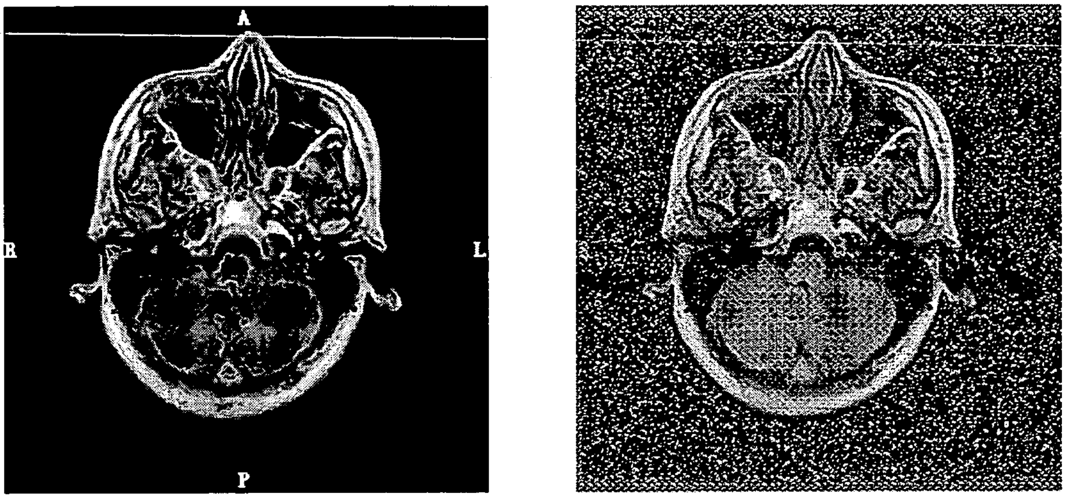
The first lossless marking technique that is suitable for high embedding rate data hiding was presented in [goljan 01]. Its main idea is as follows. The pixels in an image are divided into non-overlapped blocks, each consisting of a number of adjacent pixels. For instance, it could be a horizontal block having four consecutive pixels. A discrimination function is established to classify the blocks into three different categories, Regular, Singular and Unusable. (The authors used the discrimination function to capture the smoothness of the groups.) An invertible operation can be applied to groups. That is, it can map a gray level value to another gray level value. It is reversible since applying it to a gray level value twice produces the original gray level value. This invertible operation is hence called *flipping*. For typical images, flipping with a small amplitude will lead to an increase of the discrimination function, resulting in more Regular groups and less Singular groups. It is this bias that enables distortionless data hiding. While it is novel, and successful in distortionless data hiding, the amount of hidden data by this technique is still not large enough for certain applications. The pay-load was estimated to be in a range from 3,000 bits to 24,000 bits for a $512 \times 512 \times 8$ gray image according to [goljan 01]. Another problem with the method is that when the capacity increases, the visual quality will drop severely. For instance, PSNR drops to as low as 35 dB. Sometimes, unpleasant artifacts may occur.

The method based on the integer wavelet transform [xuan 02] is a recently proposed reversible data hiding technique that can achieve a quite large capacity. Its main idea is as follows. After the integer wavelet transform is applied to the original image, the *bias* between binary 1s and 0s in the bit-planes of the subbands LH, HL, HH is largely increased. Hence, the 1s and 0s in these bit-planes can be losslessly compressed to leave a large space for data hiding. After data embedding, inverse integer wavelet transform is applied to form the marked image. The capacity achieved in this technique is quite large. The PSNR of the marked image is, however, not high due to the histogram modification applied in order to avoid overflow/underflow. For instance, the PSNR is only 28 dB for a few images.

The method based on histogram manipulation [ni 03] is a newly invented lossless data hiding technique, which can embed a large amount of data (5k-80k bits for a $512 \times 512 \times 8$ grayscale image) while keeping the high visual quality (the PSNR is guaranteed to be above 48 dB) for a vast majority of images.

Among all of these lossless data hiding techniques, however, there is only one prior lossless data hiding technique that can be robust against compression applied to the stego-image [vleeschouwer 01]. That is, only with the technique discussed in [vleeschouwer 01] the hidden data can still be extracted out correctly after the stego-media have gone through compression within a reasonable extent. For all of the rest techniques, the hidden data cannot be recovered without error after stego-media compression.

While the above technique is robust to compression, it generates annoying salt-pepper noise because it uses modulo 256 addition. That is, when the pixel gray value is close to 256 (brightest) and/or 0 (darkest), the modulo 256 addition will likely cause flipping over between the brightest and darkest gray values. This often happens with medical images. One example is shown in Figure 1, where Figure 1 (a) is an original medical image, and Figure 1 (b) a stego-image. Obviously, this type of salt-pepper noise is annoying and not acceptable for many applications.



(a) (b)

Figure 1. (a) Original medical image and (b) stego-image with salt-pepper noise.

Summary Of The Invention

The invented method and apparatus provide a novel and more advanced lossless data hiding technique that 1) does not use modulo 256 addition, hence avoiding salt-pepper noise; 2) is more robust against stego-media compression than the only existing lossless data hiding that can resist compression [vleeschouwer 01].

Details of the Invention

The method is based on the following statistics. That is, consider a block of an image. Randomly divide this block into two sub-groups, each having the same number of pixels. Since the division is random, the statistical property of theses two sub-groups will be similar. For instance, the average gray level values of these two sub-groups will be similar. Via a different manipulation of this statistics, we invent our lossless data hiding technique that does not generate salt-pepper noise and is robust against JPEG2000 compression. It uses channel coding technique such as the BCH codes. It also uses scrambling technique.

We use "Lena" image as an example to illustrate our main idea. For a given grayscale image, say, the Lena image (512 × 512 × 8), we first split it into non-overlapping blocks. For example, the block size can be 8×8. In each block, we split it into two sub-sets as shown below, i.e., one subset consists of all pixels marked by '+', the other '-'. Each sub-set has 32 pixels. From each block, we calculate the difference value α . The difference value α is defined as the arithmetic average of differences of grayscale values of pixel pairs within the block. Specifically, one may define pixel pairs as two neighboring pixels horizontally, say, from left to right, from top to bottom, as shown in Figure 2.

+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+

Fig. 2: Difference pair pattern.

Since in a local block, the pixel values are highly correlated, the difference value α is expected to be very close to zero. The experimental results have supported this observation. The distribution of the difference value α of each block is shown in the Figure 3. It shows most values of α are very close to zero (or the mean value of this distribution is zero).

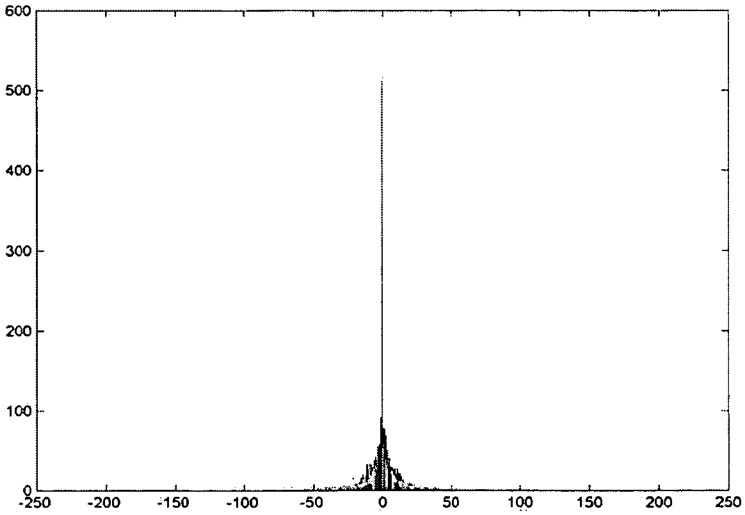


Figure 3: the distribution of the difference value α .

Since the difference value α is based on all pixels in each block, this value α has certain robustness against attacks (such as compression and other slightly alteration). We select this difference value α as a robust quantity and use it to embed data.

Data embedding strategy:

A. One bit embedding

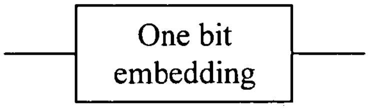


Fig. 4: one-bit embedding.

We embed one bit in each block.

1. If to be embedded bit is '1', we shift the difference value α to the right side or left side beyond a threshold, by adding or subtracting a fixed number from each pixel value within one sub-set, marked by '+'. Refer to Figure 5.
2. If to be embedded bit is '0', the pixel value of that block is intact.

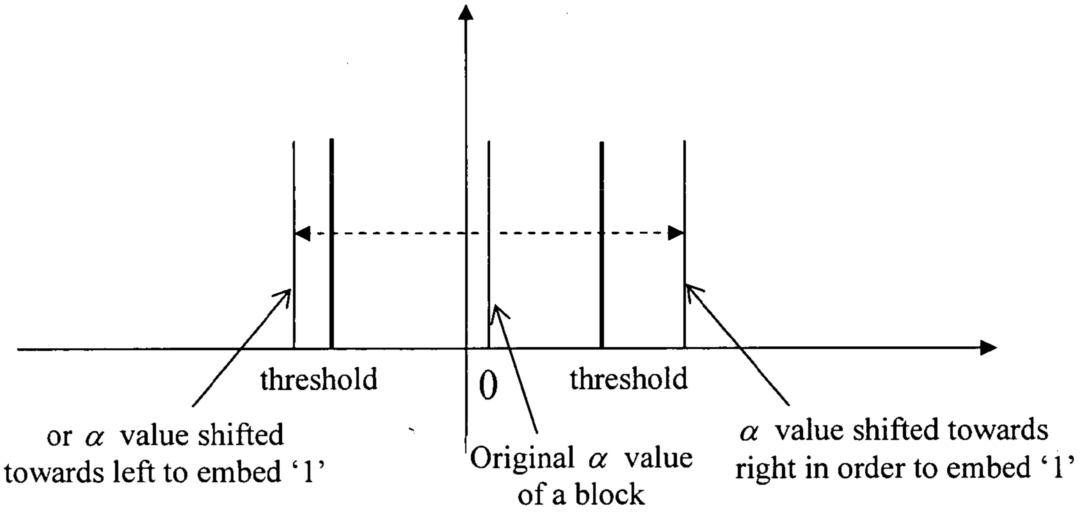

threshold
0
threshold
or α value shifted towards left to embed '1'
Original α value of a block
α value shifted towards right in order to embed '1'

Figure 5: Embedding a bit '1'.

Embedding Process

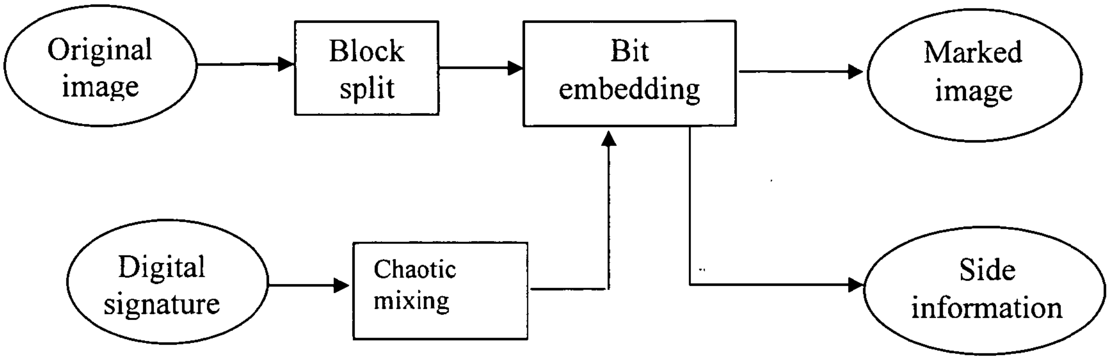


Fig. 6: Block diagram of data embedding.

1) Several parameters are considered.

   Color plane: which color plane is selected to embed

   Block size: such as block 8×8, 16×16, etc. Embed one bit in each block.

   Threshold: no large than 4 in the algorithm

   Shift quantity: two times of the threshold

   BCH (n,k,d) codes: five BCH codes for selection.

   Number of chaotic mixing operations applied: at most 30 times.

2) All the above parameters are adjusted repeatedly for better performance.
3) If after the above adjustment the BCH decoder still cannot error-freely decode embedded bits, we lower data embedding capacity.

Side information is the bits that are conveyed separately from the image itself and can be stored as image header. It includes some parameters and positions of some unusable blocks.

Overflow/underflow

In the data embedding process, we may meet the overflow/underflow problem, which will lead to truncation, hence violating lossless data hiding. If the pixel values only fall into one side of the histogram, we may shift the pixel value to the other side to avoid the overflow problem. The more difficult situation is, however, described below. Consider such a block, in which the pixel value fall into both sides of the histogram, i.e., pixels having gray scale values close to both 0 and 255. No matter we add or minus a fixed number to pixel values on one sub-set, the pixel values will beyond the range of [0,255]. Hence it may introduce permanent distortion to the image. In this case of overflow/underflow, we do nothing to this block, which means we actually embed bit '0' to this block no matter the actual to be embedded bit is '1' or '0'. In this way, it will lead to a bit error in the watermark extraction. We in fact resort to using error-correction-code (ECC) to correct this type of bit errors at the price of sacrificing the data embedding capacity.

Error correction code

In the algorithm, a few BCH codes [2] are included for selection. They are BCH (15,11,1), BCH (15,7,2), BCH (15,5,3), BCH (31,6,7), BCH (63,7,15). The reason to present these BCH codes for selection is to facilitate trade off between the coding ratio (hence correction capability) and the payload. For example, BCH (63,7,15) code is the most powerful code among these codes. It can correct 15 error bits within a codeword of 63 bits at the price of sacrificing some data embedding capacity.

Chaotic mixing

To combat the burst error, which may fail our algorithm, we introduce chaotic mixing [3] on the watermark matrix to spread the burst error evenly in the whole watermark matrix. Let $r = (x,y)$ be the location of an element in the watermark matrix, and $r'=(x',y')$ be the new location of the element in the mixed watermark matrix, we have $r' = A\ r$, where $A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ in our algorithm. We should point out that chaotic mixing is only applicable to the square matrix. But most test images are rectangular. Hence a mapping from rectangular to square is needed.

Threshold selection

In order to improve the invisibility of watermark (or PSNR between the original image and the marked image), the shift quantity should be small. Hence the threshold should also be small. In our algorithm, the threshold is at most 4. But there are some blocks, their difference values α are greater than the threshold. In order to keep the distortionless, in the data embedding, these difference values α have to be shifted by a larger quantity, i.e., further leaving away from the zero point. This means that we always embed bit '1' in these blocks no matter the actual to be embedded bit is '1' or '0'. This situation also leads to error bit and can be corrected by ECC code.

In very rare cases, the pixel values of above-mentioned blocks happen to be on both sides of the histogram. It leads to confliction. We cannot embed bit '1' or '0' on these rare blocks. We have to treat them as side information.

Extraction

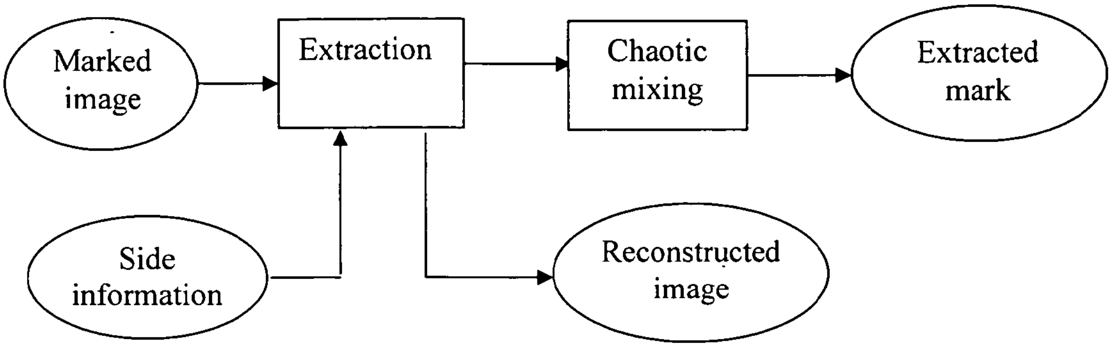


Fig. 7: Block diagram of data extraction.

Data extraction is actually the reverse process of data embedding. For a given marked image, we first split it into non-overlapping blocks and then calculate the difference value α for in each block in the same way as that in data embedding.

1. If the difference value α is beyond the threshold, then bit '1' is extracted and the difference value is shifted back to its original position, which means the pixel value of one sub-set is back to its original value.
2. If the difference value α is below the threshold, then bit '0' is extracted and nothing to do on the pixel value of that block.

In this way, we can extract the watermark and get the original image without any distortion.

AUTHENTICATION

Content signing

In our algorithm, the embedded data is a digital signature produced from the content feature. We first extract the content feature from the image, then use one way hash and private/public key encryption to get the digital signature. The length is 1024 bits or 512 bits. The process is shown in Fig. 8.

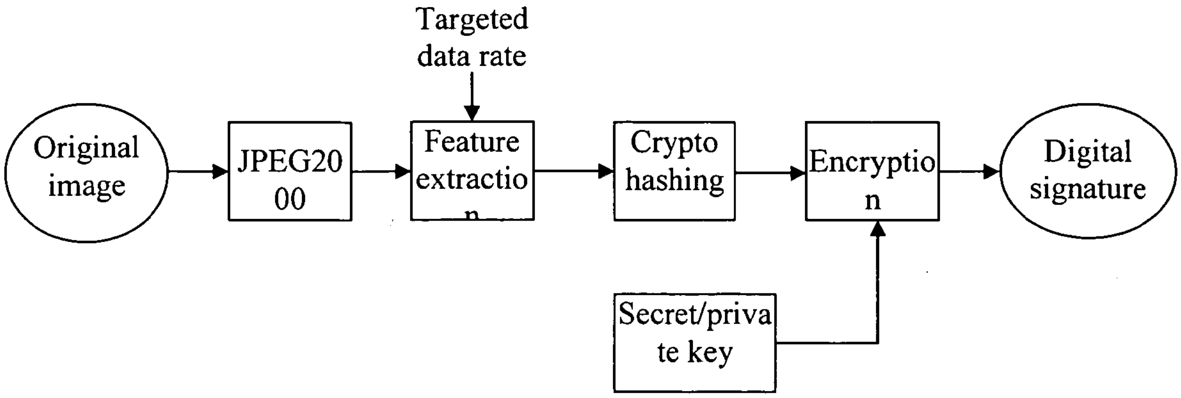


Fig. 8: Block diagram of content signing.

Signature embedding

The digital signature is embedded into the original image according to the lossless data hiding algorithm in section II to get the watermarked image.

Authentication

The whole authentication process is shown in Fig. 9. The extracted mark and the reconstructed image are gotten from the watermarked image according to the extraction technique in section II.

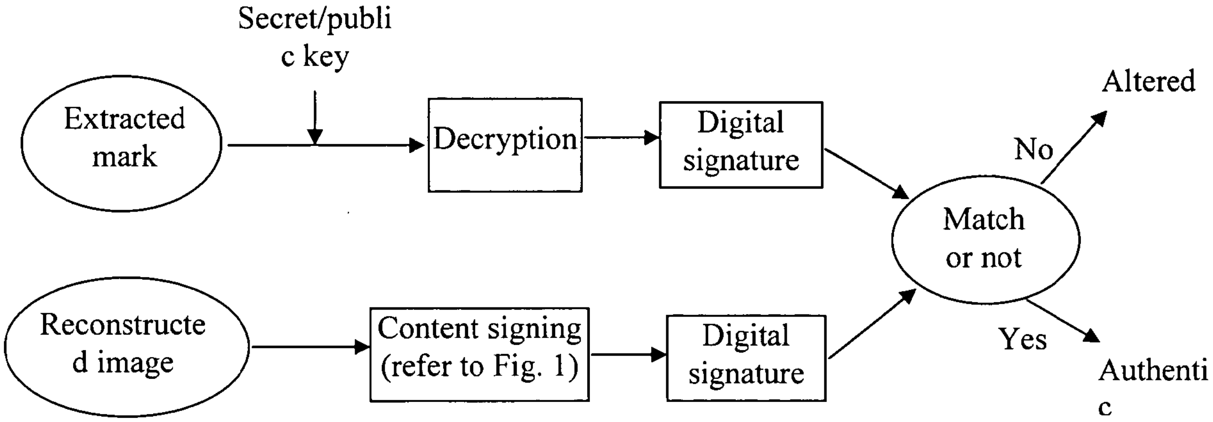


Fig. 9: Block diagram of authentication.

Localization: the authentication process can also be used to check which local part of the image has been changed if the image has been altered. If the local extracted bit does not match the produced signature bit, it shows that block has been changed.

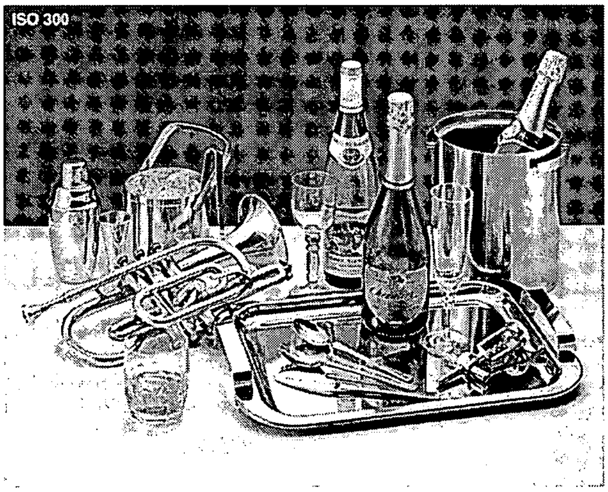
Our novel robust distortionless image data hiding technique has certain robustness to JPEG and JPEG2000 compression. This technique has a few advantages over the existing robust distortionless data hiding technique. These advantages are as follows: No salt-and-pepper noise at all. Applicable to all commonly used image (including medical image, more than 1000 images in the database of CorelDRAW and all JPEG2000 test images). Average PSNR of marked images is above 37 dB. Robust to JPEG2000 or JPEG compression to a certain extent. Data embedding capacity is 1024 bits or 512 bits for JPEG2000 test images. Used for image authentication and semi-robust integrity verification

Experimental Results

We have successfully applied our proposed algorithm to some commonly used grayscales images such as 'lena', 'baboon', etc., some medical image and more than 1000 images in the CoralDraw image database.

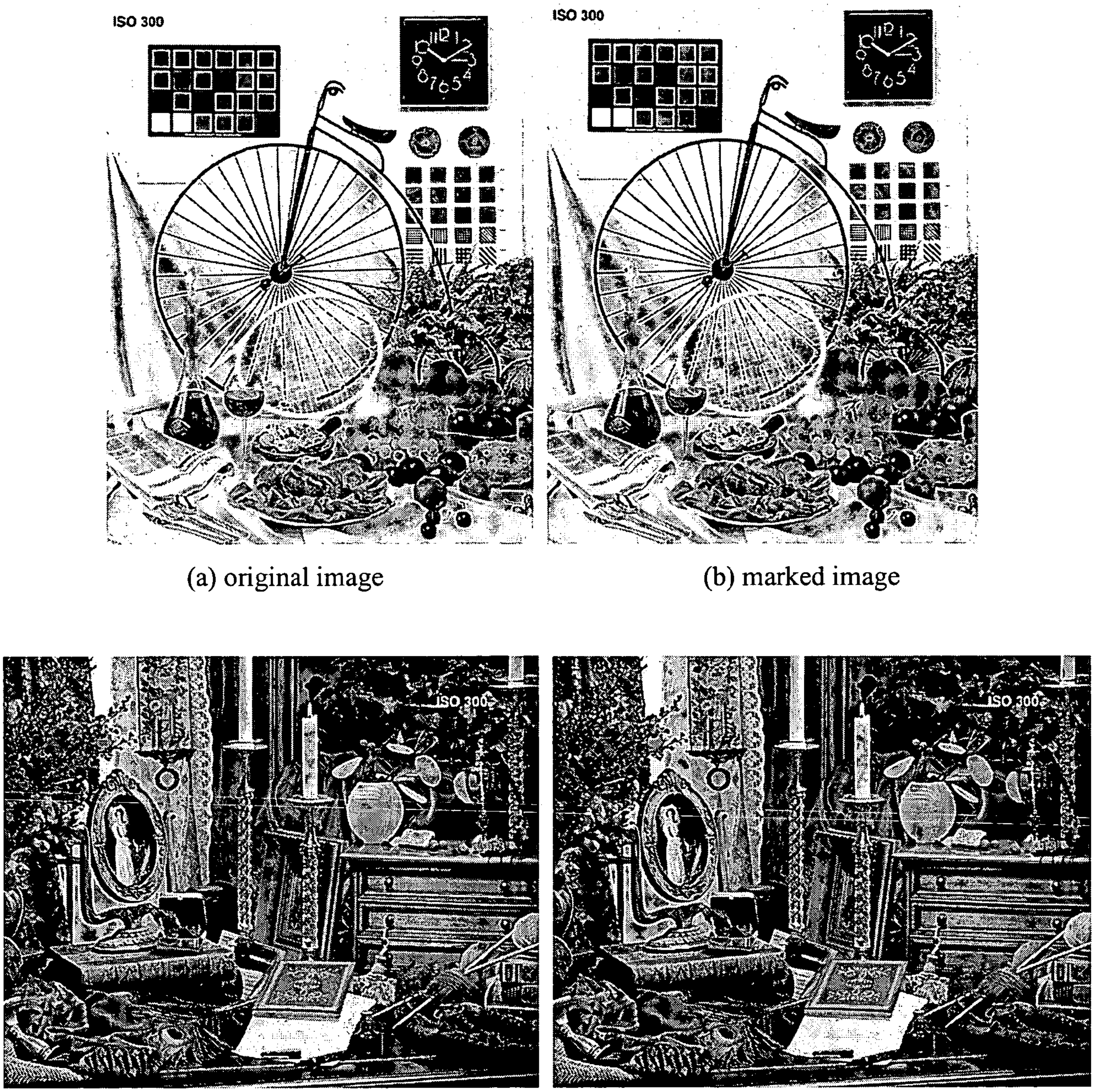
It is noted that there is no salt-and-pepper noise at all since we do not use modulo 256 addition in our algorithm. The embedding capacity is above 1024 or 512 bits. The average PSNR is above 37 dB. They can resist the JPEG2000 compression attack from 2.0 bpp to 0.2 bpp.

Further more, we have successfully applied our algorithm to eight JPEG2000 test images. It shows that our algorithm can be applied to *all* these test images. The follows are some experimental results.



(a) original image

(b) marked image



(a) original image (b) marked image

(a) original image (b) marked image

Fig. 10: Some pairs of original and marked JPEG2000 test images.

Table 1 is the experimental results of eight JPEG2000 tested image. Among them, the robust means at what level JPEG2000 lossy compression (bit per pixel), the embedded data still can be extracted without any error. The side information includes some parameters used in the watermark extraction and the coordinates of some unused blocks. The side information are put outside the image.

Table 1: experimental results for eight JPEG2000 test images.

Images (1536x1920)	PSNR of marked image (dB)	Data embedding capacity (bits)	Robustness (bpp)	Side information (bits)
N1A	45.1	1398	0.8	120
N2A	43.1	1398	1.6	490
N3A	45.1	1398	1	170
N4A	45.2	1398	1	480
N5A	45.5	1200	1	380
N6A	45.0	1267	0.4	210
N7A	40.6	1398	1.2	410
N8A	41.5	798	1.4	770

Table 2 and table 3 are the experimental results for eight medical images and eighty CorelDraw images, respectively.

Table 2: experimental results for eight medical images.

Images (512x512)	PSNR of marked image (dB)	Data embedding capacity (bits)	Robustness (bpp)
Mpic1	40.3	476	0.8
Mpic2	36.4	476	0.8
Mpic3	41.4	476	0.6
Mpic4	41.3	476	1
Mpic5	39.8	476	0.8
Mpic6	36.4	476	0.8
Mpic7	40.2	476	0.4
Mpic8	40.9	476	0.8

Table 3: experimental results for eighty CorelDraw images.

Images (512x768)	PSNR of marked image (dB)			Data embedding capacity (bits)	Robustness (bpp)		
	Max	Min	Avg	476	Max	Min	Avg
	45.2	35.4	38.4		2.2	0.4	1.47

Additional Experimental Results

The method has been applied to more than 100 frequently used images. The five original and marked image pairs are shown below (Fig. 2-6). The test results are listed in Table 1. Clearly, there is no annoying salt-pepper noise even for medical images. It is also shown that the technique is robust against JPEG compression. There are about 80 images among the 1000 images in the database of commercial software CorelDRAW, to which the lossless data hiding technique in [vleeschouwer 01] could not be successfully applied [zou 03]. The proposed technique has been successfully applied to all of these 80 images. That is, our invented method can utilize the statistics mentioned above more effectively than the method reported in [vleeschouwer 01].

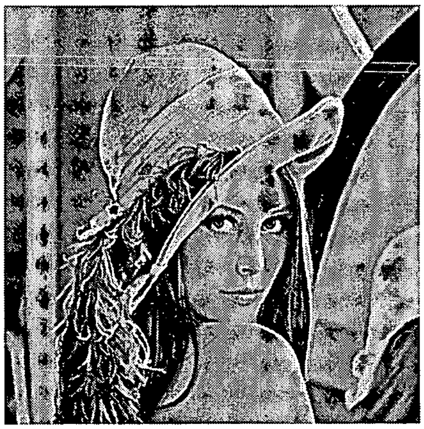


Figure 2 (a) Original Lena image.

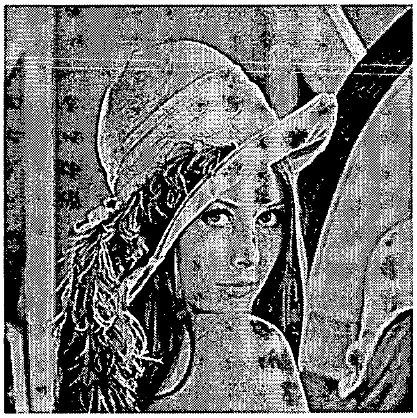


Figure 2 (b) Marked Lena image.

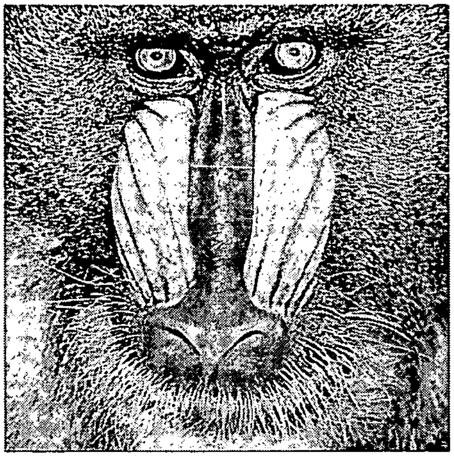


Figure 3 (a) Original Baboon image.

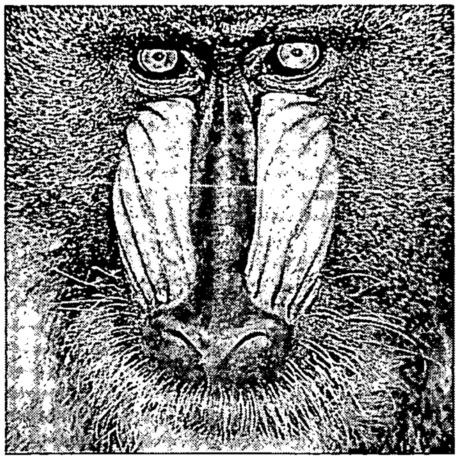


Figure 3 (b) Marked Baboon image.

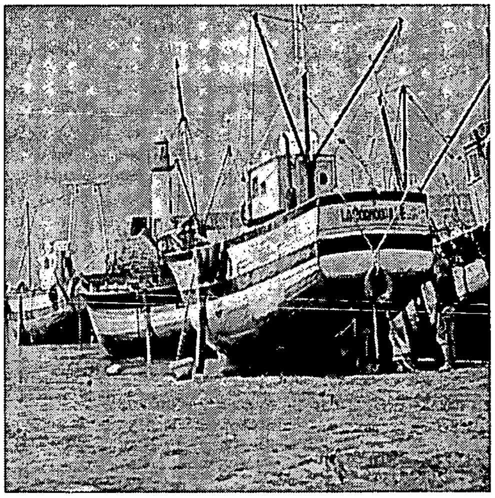


Figure 4. (a) Original Boat image.

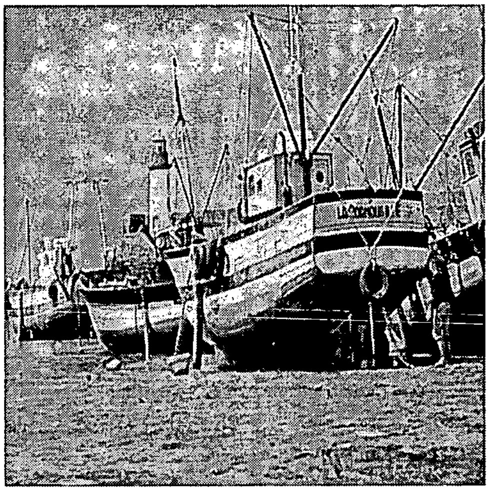


Figure 4 (b) Marked Boat image.

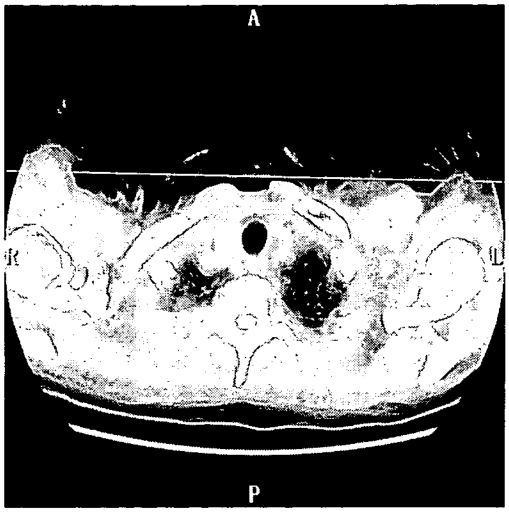


Figure 5 (a) Original medical image 1.

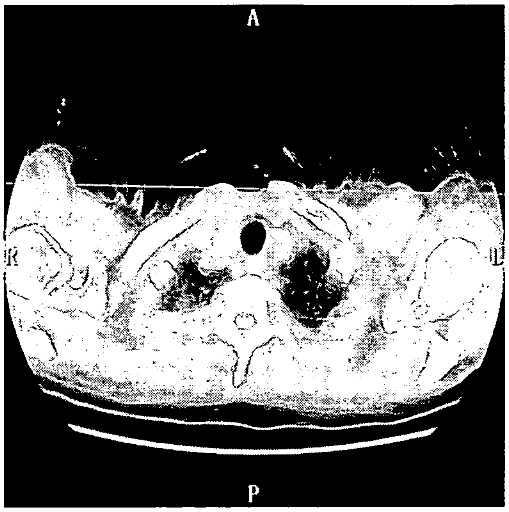


Figure 5 (b) Marked medical image 1.

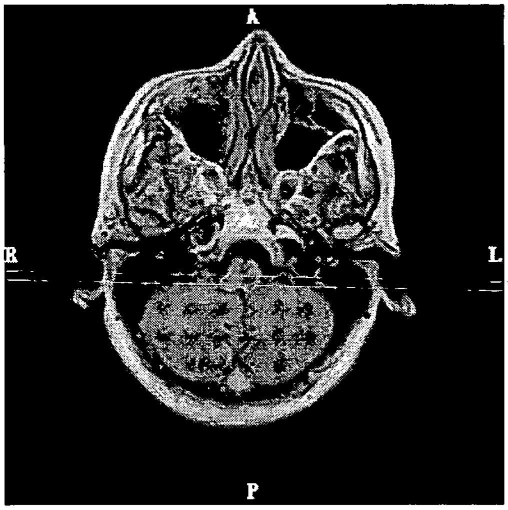


Figure 6 (a) Original medical image 2.

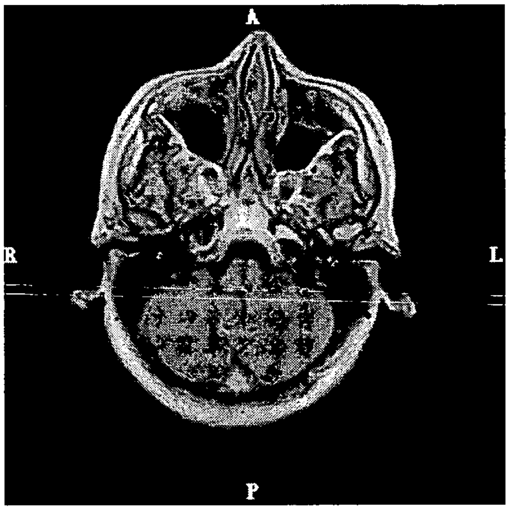


Figure 6 (b) Marked medical image 2.

Table 1 Some experimental results (500 information bits embedded in 512*x*512 gray images, bpp denotes bits per pixel).

Images (512x512x8)	PSNR of marked image (dB)	Data rate survived (bbp)
Lena	44.7	1.2
Lena	42	1.0
Lena	38	0.6
Baboon	37.3	2.0
Boat	38.6	0.8
Medical image 1	34.3	0.24
Medical image 2	44.7	0.64

Claims

1. Methods and apparatus as described herein.

REFERENCES

[fridrich 01] J. Fridrich, M. Goljan and R. Du, "Invertible authentication," *Proc. SPIE, Security and Watermarking of Multimedia Contents*, San Jose, CA, January (2001)

[goljan 01] M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding," *Proceedings of 4th Information Hiding Workshop*, Pittsburgh, PA, April, 2001.

[honsinger 99] C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," US Patent application, Docket No: 77102/E–D (1999)

[macq 99] B. Macq and F. Deweyand, "Trusted headers for medical images," *DFG VIII-D II Watermarking Workshop*, Erlangen, Germany, Oct. 1999.

[vleeschouwer 01] C. de Vleeschouwer, J. F. Delaigle and B. Macq, "Circular interpretation on histogram for reversible watermarking," *IEEE International Multimedia Signal Processing Workshop*, Cannes, France, pp.345-350, October 2001.

[xuan 02] Guorong Xuan, Jidong Chen, Jiang Zhu, Yun Q. Shi, Zhicheng Ni, Wei Su, "Distortionless Data Hiding Based on Integer Wavelet Transform," *IEEE International Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin islands, December 2002.

[ni 03] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE International Symposium on Circuits and Systems*, May 2003, Bangkok, Thailand.

[zou 03] D. Zou, C. W. Wu, G. Xuan and Y. Q. Shi, "A content-based image authentication system with lossless data hiding," *IEEE International Conference on Multimedia and Expo*, July 2003, Baltimore, Maryland.

[1] C. De Vleeschouwer, J.F. Delaigle and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management" *IEEE Transactions on Multimedia,* March 2003.

[2] J. G. Prokis, "Digital Communications," fourth edition, McGraw-Hill, 2000.

[3] G. Voyatzis and I. Pitas, "Chaotic mixing of digital images and applications to watermarking," in *European Conference of Multimedia Applications, Services Techniques (ECMAST'96)*, 2, pp. 687-695 (May 1996).

Appendix A

What follows are the comparison results between the Vleeschouwer et al's method and our proposed method. It shows that the PSNR of our method is much larger than that of Vleeschouwer's method and there is no any salt-pepper noise in our method. It also shows that the robustness of our method is better than that of Vleeschouwer's method.

		Eight Medical images (512x512x8, 476 Information bits embedded)	
		De Vleeschouwer's	Our method
PSNR (dB)	max	26.49	41.35
	min	4.73	36.36
	avg.	14.31	39.60
Robustness (bpp)	max	2.8	1.0
	min	0.6	0.4
	avg.	1.27	0.75
Salt-pepper noise		Yes	No

		80 CorelDRAW images (512x768x8, 714 Information bits embedded)	
		De Vleeschouwer's	Our method
PSNR (dB)	max	30.06	45.23
	min	5.27	35.36
	avg.	21.97	38.42
Robustness (bpp)	max	3.0	2.2
	min	0.6	0.4
	avg.	1.40	1.47
Salt-pepper noise		70% a little 8% severe	No

		Eight JP2000 Test images (1536x1920x24, 1412 Information bits embedded)	
		De Vleeschouwer's	Our method
PSNR (dB)	max	25.39	45.21
	min	14.27	41.34
	avg.	19.92	43.63

Robustness (bpp)	Max	3.0	1.6
	Min	0.6	0.4
	Avg.	1.62	0.95
Salt-pepper noise		Yes	No

Appendix B

What follows is a technical paper that provides additional information related to our invention above. It is our understanding that as of this filing, this paper has not been published or otherwise disclosed publicly.

ISO/IEC JTC 1/SC
29/WG 1
(ITU-SG8)

Coding of Still Pictures

JBIG
JPEG
Joint Bi-level Image
Joint Photographic
Experts Group
Experts Group

TITLE:	**A Unified Authentication Framework for JPEG2000: Technical Description**
SOURCE:	Zhishou Zhang, Gang Qiu, Qibin Sun and Xiao Lin (Institute for Infocomm Research, Singapore) Zhicheng Ni and Yun-Qing Shi (New Jersey Institute of Technology, USA)
PROJECT:	JPEG-2000 Part-8 (JPSEC)
STATUS:	Proposal

Table of Contents

Introduction

Traditional digital signature techniques (e.g., DSA or RSA) provide an effective and secure solution for data authentication, which covers both integrity protection and non-repudiation. Any one-bit modification will make the protected data unauthentic, which is definitely advantageous for data as every bit of data is vital. For example, if a transaction is made on-line, the exchanged data may contain information like amount of payment, account number and payee's name. As you can imagine, any modification, even a single-bit, will lead to total failure of the transaction.

Directly applying traditional digital signature techniques on image can also provide a good protection of image data, but in an unreasonably strict way. Such authentication on image is called fragile authentication. As images are exchanged between different entities within different media, they are unavoidably experiencing incidental distortion introduced by image transcoding, unreliable carrier and multi cycles of encoding-decoding. Although the incidental distortion changes image data, it doesn't change meaning of the image from human's point of view. An unattacked image that experienced incidental distortion will be declared as unauthentic with traditional digital signature based authentication scheme. Therefore, the fragility of traditional digital signature techniques limits the typical applications of image.

JPEG2000 has many advanced features, including lossy-to-lossless compression, better compression ratio, resolution scalability, quality scalability, ROI and so on. Therefore we should have the following principles in our mind when we design an authentication system for JPEG2000. They are:

- The authentication framework must be able to exploit advanced features of JPEG2000. For instance, the solution should be able to protect the JPEG2000 image in a scalable way. To align with JPEG2000, the solution must be able to protect any one or more components, tiles, resolutions, quality layers, ROIs, precincts, or code blocks.
- The authentication framework should be able to provide effective and secure protection for JPEG2000 images, while being robust enough for incidental distortion.
- The authentication framework should not be obtained in a way either by compromising those advanced features of JPEG2000 or by narrowing its typical applications. For instance, the solution should retain the lossless feature of JPEG2000 image.
- The authentication framework should be compatible with state-of-arts information security standards such as X.509, etc.

In the document N2946 and N3074, we have already proposed a unified content-based authentication framework for JPEG2000 images, and have given system description and test results. This document is to give more technical details of our proposed authentication system.

Terms and Definitions

Authentication Authentication is the process to protect integrity of data and to prevent repudiation. Usually It involves signing process and verification process.

Digital Signature Digital signature is a natural tool for data authentication. General speaking, it should comprise some data (i.e., the signature) that the receiver can keep as evidence that a particular message was sent and that the signer was the originator.

Sign Sign is the process of generating a signature for the protected data.

Verify Verification is the process of detecting any possible attacks from the protected data.

Hash It means one-way hash function in cryptography. Typical hash functions are MD-5 and SHA-1.

Lowest Authentication Bit Rate (LABR) It refers to the authentication strength. As long as the bit-rate of the re-coded or transcoded JPEG2000 is greater than the LABR, its authenticity is guaranteed by our proposed solution.

Fragile Authentication With Fragile Authentication, protection is based on image data instead of image content. Any even one-bit modification within the protected image data will make the image unauthentic, even if it doesn't change its content meaning.

Lossy Authentication With Lossy Authentication, protection is based on image content. Lossy Authentication is robust against the defined incidental distortions by watermarking the content in a lossy way.

Lossless Authentication With lossless authentication, protection is also based on image content, and it's also robust against the defined incidental distortions. But it can recover the original image after watermark extraction, if no transcoding is applied.

Incidental Distortion Incidental distortion is introduced by common image processing and unreliable network transportation. Normally, incidental distortion doesn't change the image meaning but degrade the image quality.

Intentional Distortion Intentional distortion is introduced by some kind of malicious attack, which changes the meaning of image content.

Lossy watermarking Watermarking will permanently cause the degradation of image quality, though it is imperceptible.

Lossless watermarking Watermarking will cause the degradation of image quality, though it is imperceptible. However, the original content can be exactly recovered after watermark extraction, if no transcoding is applied on the watermarked JPEG2000 image.

Error Correction Coding (ECC) A coding system that incorporates extra parity bits in order to detect and correct errors.

Parity Check Bits (PCB) The redundancy bits in order to detect and correct errors.

Attack In robust authentication (lossy or lossless), it refers to any content modifications which result in a change of content meaning. In fragile authentication, any one-bit change will be deemed as attack.

* Note that the terms and definitions list here are just for the purpose of understanding this document and may not be the same as their formal ones.

Functionalities

The proposed system integrates fragile authentication, lossy authentication and lossless authentication in one single unified framework for JPEG2000 images. Similar to JPEG2000 compression strength that is quantitatively controlled by the compression bit-rate, the authentication strength could also be quantitatively specified by a parameter called "Lowest Authentication Bit Rate (LABR)". It means that all data/content of JPEG2000 image above LABR will be protected. Thus it will bring users much convenience.

Fragile authentication is used to protect one or more parts of codestream, or even the whole codestream from the main header to EOC marker. Since it's fragile, any one-bit modification of the protected part will make the image unauthentic. Lossy authentication is used to protect JPEG2000 image in a semi-fragile way, which is much more robust to incidental distortion. The image quality after lossy authentication degrades in an imperceptible way, due to watermark embedding. Similarly, lossless authentication also protect JPEG2000 image in semi-fragile way, but the original image can be recovered after watermark extraction, assuming no transcoding is applied. Typical functionalities of the proposed system are listed below.

Fragile Authentication

In fragile authentication mode, JPEG2000 image can be protected in various granularities, including the following:

- Protect the whole code stream.
- Protect part of the code stream pertaining to one or more tiles.
- Protect part of the code stream pertaining to one or more components.
- Protect part of the code stream pertaining to one or more resolution levels.
- Protect part of the code stream pertaining to one or more quality layers, defined by LABR.
- Protect part of the code stream pertaining to one or more precincts.
- Protect part of the code stream pertaining to one or more code blocks.
- Protect part of the code stream pertaining to one ROI.

Lossy Authentication

With Lossy authentication, the signature can survive incidental distortion such as transcoding and multi cycles of JPEG2000 encoding-decoding. However, if image content is intentionally modified, i.e. content meaning is changed, it will not be able to pass the verification process. As indicated by its name, it is lossy in the sense that image quality imperceptibly degrades after watermark embedding.

Similarly, the image can be protected in the following granularities:

- Protect the whole image content
- Protect image content of one or more quality layers, defined by LABR.
- Protect image content of one or more tiles.
- Protect image content of one or more components.
- Protect image content of one or more ROIs.
- Protect image content of one or more resolutions.
- Protect image content of one or more precincts.
- Protect image content of one or more code blocks.

In addition, with lossy authentication, it is able to allocate the attacked area, should the image be maliciously manipulated.

Lossless Authentication

Lossless authentication goes one step further. It could recover the original image after watermark extraction (if no any transcoding is applied). If transcoding is applied, the original may not be recovered. However, transcoded image can still be verified as authentic so long as the bit rate of the transcoded image is above the LABR. (It also provides the robustness against these incidental distortions). With lossless authentication, it is also able allocate the attacked area.

The image can be protected in the following granularities:

- Protect the whole image content.
- Protect image content of one or more quality layers, defined by LABR.
- Protect image content of one or more tiles.
- Protect image content of one or more components.
- Protect image content of one or more ROIs.
- Protect image content of one or more resolutions.
- Protect image content of one or more precincts.
- Protect image content of one or more code blocks.

General Description

Figure 1 give an illustration of the proposed system for JPEG2000 image authentication. The left part is the encoder and the right part is the decoder. The encoder accepts three sets of parameters, including encoding parameters (such as CBR, 5/3 filter or 9/7 filter, etc), original image to be encoded and authentication parameters (such as LABR, protected locations and authentication mode). Depending on the specified authentication mode, different authentication module will be invoked while the image is being encoded. If fragile authentication is specified, the "fragile sign" module is invoked to generate the signature, which is a straightforward solution with traditional crypto signature; If lossy authentication is specified, the "lossy sign" module is invoked to embed watermark into the image and generate signature, which is supposed to be more robust to incidental distortions; If lossless authentication is specified, the "lossless sign" module is invoked to embed watermark into the image and generate signature, such that after signature verification, the image content can be exactly recovered if no transcoding is applied. If transcoding has been applied to the image, the JPEG2000 image can still be verified but cannot be exactly recovered. The final outputs are a JPEG2000 image (without watermark for fragile authentication and with watermark for lossy & lossless authentication) and its associated digital signature.

In the reverse direction, a decoder accepts four inputs: JPEG2000 image to be decoded, digital signature, public key and authentication parameters. Similarly, depending on the specified authentication mode, different verify module (fragile verify, lossy verify or lossless verify) will be invoked while the image is being decoded. The final outputs of the decoder are the decoded image, verification status and information about the attacked areas (in case that the image is maliciously manipulated). Note that after lossless verification, the decoded image will be exactly the same as the original image.

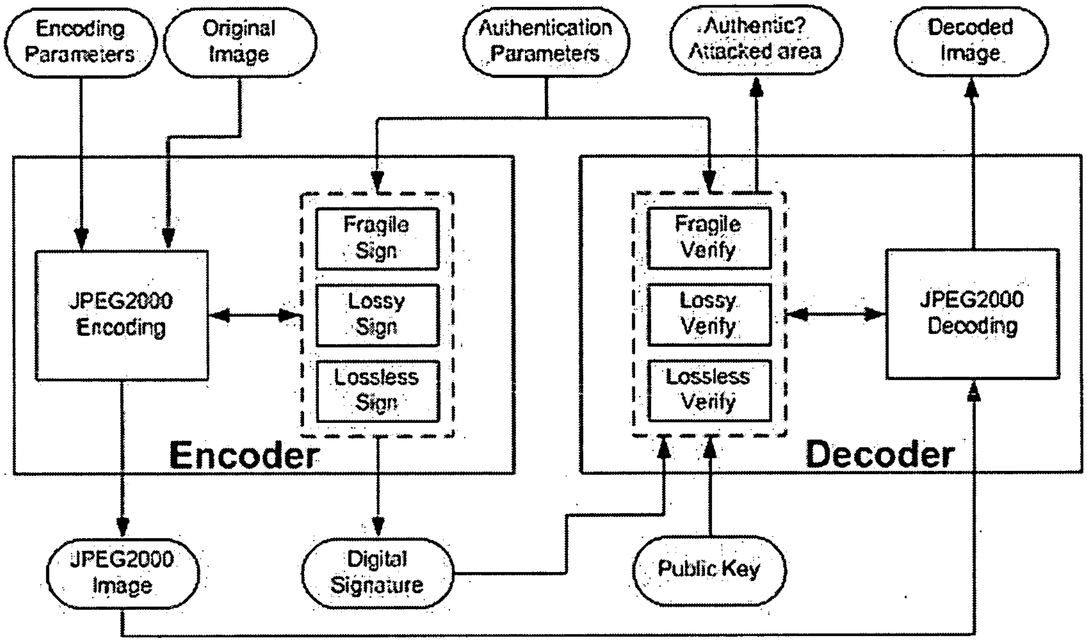


Figure 1. the diagram of proposed solution

Fragile Authentication

Fragile authentication is selected for protecting JPEG2000 image on code-streams level. Fragile signing and verifying operations are quite straightforward, as shown in Figure 2 and 3. During sign operation, the original image is encoded as per normal. While the code stream is being formulated, its protected parts, as specified by LABR and other parameters, are extracted and fed to traditional hashing and signing operation. As result, a digital signature is generated. During verify operation, while the code stream is parsed during decoding, its protected part, as specified by LABR and other parameters, is extracted and fed to traditional hashing and verifying operation, which returns the verification result. Even one-bit change in the protected part will be deemed as unauthentic.

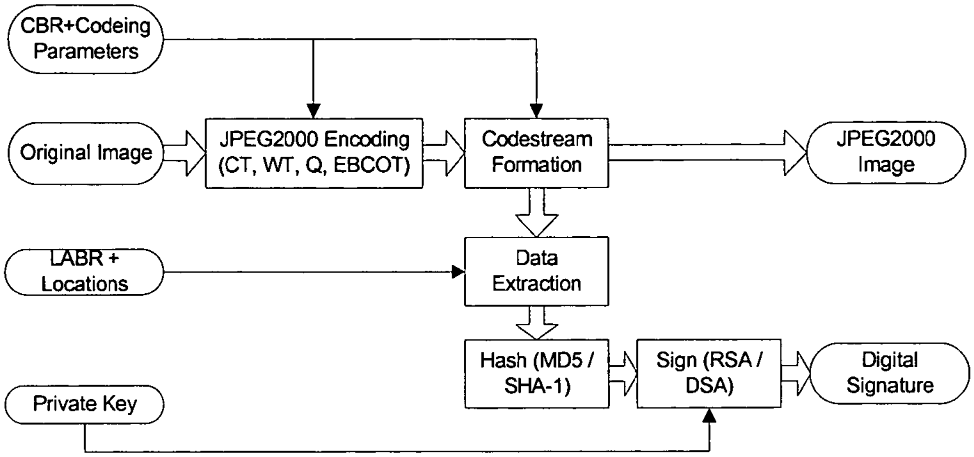


Figure 2. Fragile sign operation

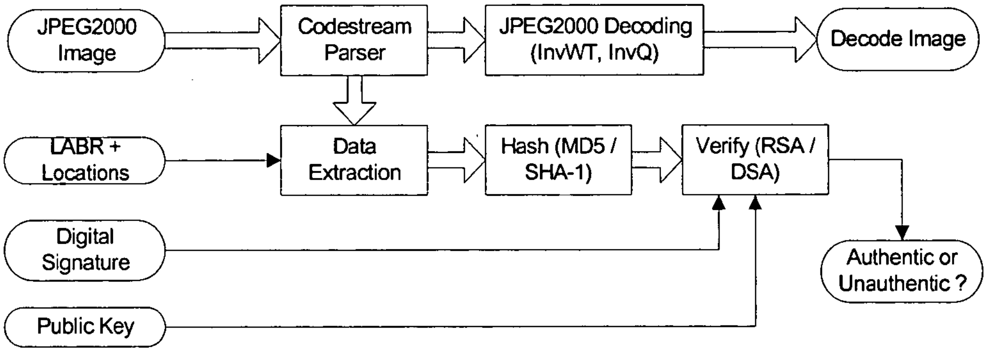


Figure 3. Fragile verify operation

Lossy Authentication

Lossy authentication is usually selected for those applications demanding for more robustness such as wireless communication. Figure 4 illustrates the basic ideas of lossy signing operation. Firstly, the original image undergoes color and wavelet transformation, quantization, arithmetic coding and EBCOT, which are all basic procedures in JPEG2000 encoding. EBCOT process will find out for each coded block those bit-planes that are above LABR (i.e., they survive transcoding operation to LABR). Then, decision is made on which resolution level (X) is suitable for feature extraction and which resolution level (Y) for watermark embedding, based on Human Vision System (HVS). The block-based feature, F_i, is then encoded with selected Error Correction Coding (ECC) Scheme to generate codeword CW_i. The Parity Check Bits of CW_i, PCB_i, is used as a seed to formulate block based watermark W_i, which is then embedded into the corresponding block in LH or HH subband of Y. In addition, features from all blocks are concatenated and the resulted bit sequence is hashed by a cryptographic hashing function such as MD5 or SHA-1. The generated hash value can then be signed using the content sender's private key to form the crypto signature.

Figure 5 illustrates the lossy verifying operation. The codestream parser finds out for each block those bit-planes above LABR, based on which we can decide the resolution level X for feature extraction and resolution Y for watermark extraction. Block-based feature extraction is the same to that in sign operation. Block-based watermark is extracted from each block in resolution Y. Note that if the input image is not JPEG2000 format, we have to repeat the operation that is the same as the signing to obtain the watermark and the features. Then combining features and PCBs from each block forms codeword, and the whole verification decision could be made orderly. Firstly, we calculate the syndrome of the codeword for each block to see whether any blocks are uncorrectable. If yes, then we claim the image is unauthentic and those blocks with uncorrectable codewords are attacked area. However, if all codewords are correctable (i.e. errors in any feature code are correctable by its PCB), all corrected codewords are concatenated into a bit sequence, which is then cryptographically hashed. The final verification result is concluded through a cryptographic verifying operation using supplied signature and public key.

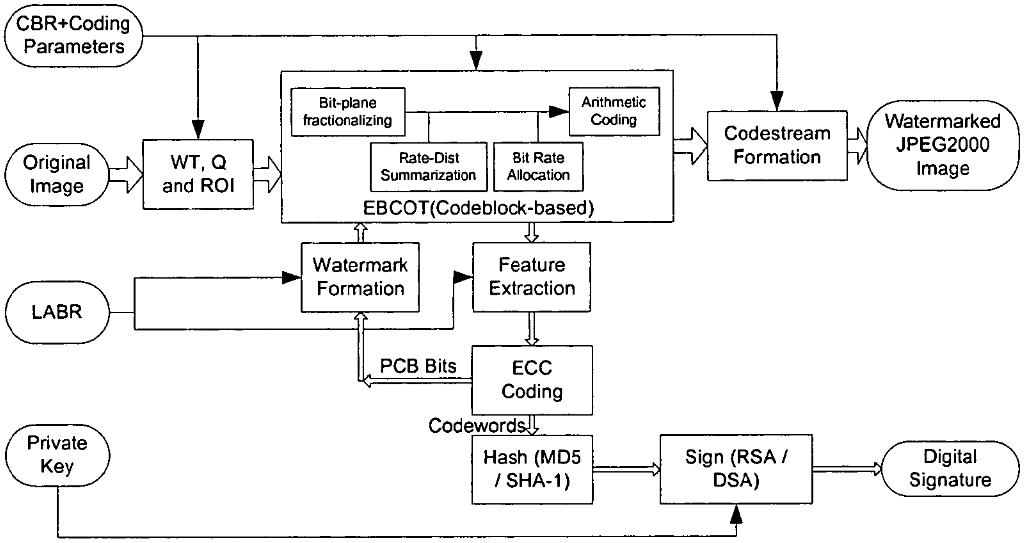


Figure 4. Lossy signing operation

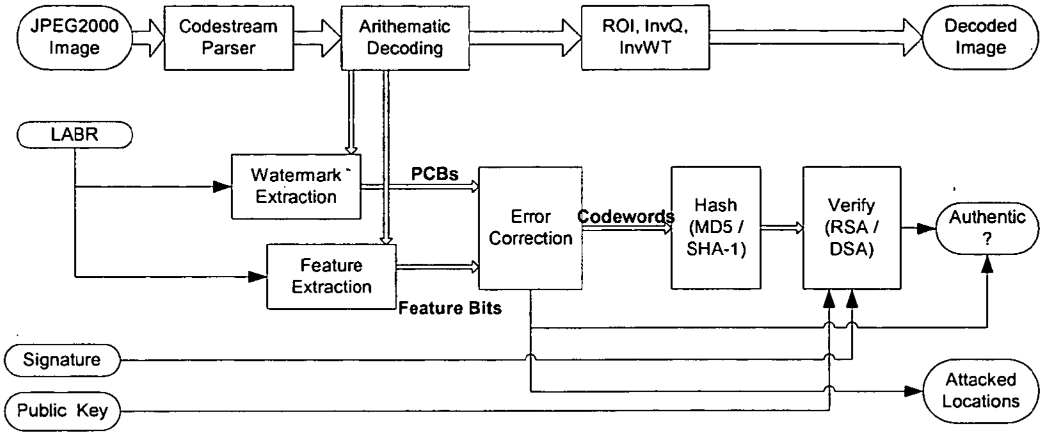


Figure 5. Lossy verifying operation for JPEG2000 file

Lossless authentication

Lossless mode is usually selected for medical or remote imaging related applications where lossless recovery of the watermarked image is required. Lossless signing operation is very similar to lossy signing operation (Figure 4). The only difference lies in watermark embedding module. The codeblock whose size is usually 64x64 is further divided into 8x8 blocks called patches. The coefficients in a patch are split into two subsets. Then we calculate the difference value α, which is defined as the arithmetic average of differences of coefficients in two respective subsets. Since in a patch, the coefficients are highly correlated, the difference value α is expected to be very close to zero. Furthermore, it has certain robustness against incidental distortions because α is based on all coefficients in the patch. Each patch is embedded with one bit, as illustrated in Figure 6. If 1 is to be embedded, we shift difference value α to right side or left side beyond a threshold, by adding or subtracting a fixed number from each coefficients within one subset. If 0 is to be embedded, the patch is intact. There are chances that the value α is originally beyond the threshold and a bit of 0 is to be embedded. In this case, we shift the value α further away beyond the threshold, and rely on ECC to correct the bit error, because the watermark bits are ECC encoded again before being embedded.

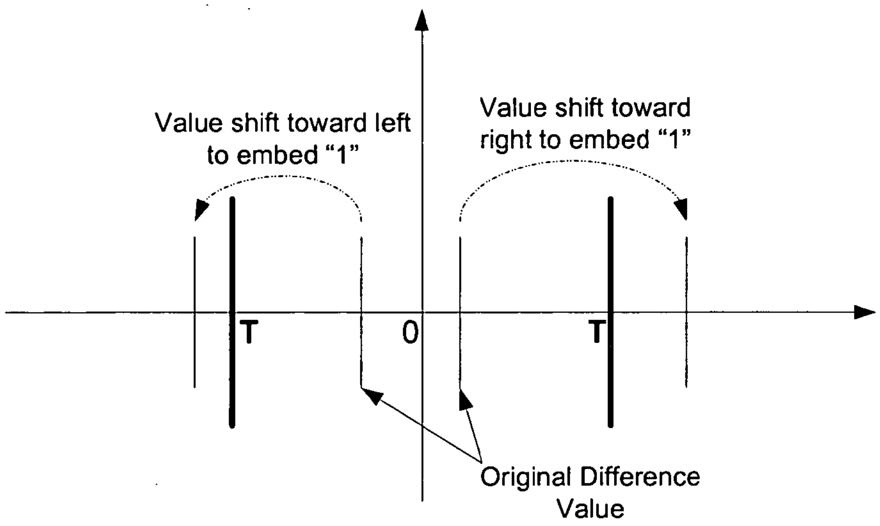

Value shift toward left to embed "1"
Value shift toward right to embed "1"
T
0
T
Original Difference Value

Figure 6. Embedding a bit "1"

Lossless verifying operation is also similar to lossy one, with the exception of watermark extraction. The code block is divided in patches and difference value α of each patch is calculated in the same way as lossless sign. For each patch, if value α is beyond the threshold, a bit of "1" is extracted and the difference value is shifted back to its original position, which means that original coefficients are recovered. If the value α is inside the threshold, a bit of "0" is extracted and nothing needs to be done. Finally an ECC correction is applied on the extracted bit sequence to get the correct watermark bits.

System Analysis

This section elaborates more on algorithmic complexity, storage overhead, data expansion, impact of transmission error and security analysis.

Complexity

For fragile sign and verify operation, the processing overhead is around 5% of standard JPEG2000 encoding/decoding time. It mainly comes from:

- Find from the code stream the protected segment that is specified by LABR and Location parameters, and extract it from the code stream. This can be done while the code stream is being formulated, thus the processing overhead is quite minimal.
- Perform one hash (MD5/SHA-1) operation on the extracted part of the code stream.
- Perform one RSA/DSA sign or verify operation.

For lossy and lossless operation, the process overhead is around 20% of standard JPEG2000 encoding/decoding processing time. It mainly comes from:

- Find out all bit-planes above LABR, in order to decide where to extract feature and where to embed / extract watermark.
- Extract features from each protected code block.
- ECC coding or correction for each protected code block.
- Watermark embedding or extraction
- One hash operation.
- One RSA/DSA sign or verify operation

Storage overhead

For fragile sign and verify, the only overhead comes from the fact that the protected part of code stream need to be temporarily stored in memory in order to sign/verify. For example, for 500KB image, the maximum memory overhead is 500KB.

For lossy and lossless authentication, the quantized coefficients in current tile need to be temporarily stored in memory in order not to repeat wavelet and quantization steps. In this case, the overhead depends on size of a tile. E.g., if tile size is 256x256, the overhead will be around 1 MB.

Data expansion

For fragile authentication, the overhead is only side information, like signature, protected location, public key and so on. It is roughly about 300 bytes. The resulted code stream is exactly same as normal encoding.

For lossy and lossless authentication, the overhead of side information is the same as fragile authentication. The resulted code stream size is about 0~200 bytes more or less than normal encoded code stream.

Impact of transmission error

For fragile authentication, any transmission error will resulted in failure of verification, due to the nature of traditional crypto signature.

For lossy and lossless authentication, as long as the number of error bits is not significant, our solution can still authenticate the image, due to the robustness of our solution.

Security Analysis

For Fragile authentication, the security strength is the same as that of the underlying Hash (MD5 or SHA-1) and Sign (RSA or DSA) algorithm.

However, for lossy and lossless authentication, content-based feature extraction and error correction coding (ECC) reduce the security strength, as some modification may not affect the extracted features or modified features can be corrected by ECC. However, such security risk can be compensated from image contextual characteristics.

Summary

We proposed a systematic and quantitative way for authenticating multimedia content by casting the content into a finer representation in terms of authentication bit rate. This then brings much convenience for the authentication applications by simply keying in one parameter—authentication bit-rate to protect the content.

We also proposed a framework for meeting different authentication requirements from real applications by employing different signing modules (fragile, lossless and lossy) which is in line with different JPEG2000 coding settings. The proposed scheme is fully compatible with JPEG2000 coding and traditional crypto schemes.

We believe that the proposed scheme is well suited to and needed by JPSEC tools.

Reference

- Zhishou Zhang, Gang Qiu, Qibin Sun, Xiao Lin, Zhicheng Ni, Yun-Qing Shi, WG1N3074 "A Unified Authentication Framework for JPEG2000 images: System Description and Experiment Results"

- Qibin Sun, Xiao Lin and Yun-Qing Shi, WG1N2946 "A Unified Authentication Framework for JPEG2000 images"

- Touradj Ebrahimi and Claude Rollin, , WG1N30555 "JPSEC Working Draft – Version 2.0"